UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/045,307	01/14/2002	Jimmie Earl DeWitt JR.	AUS920010716US1	1331

35525 7590 05/03/2007

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
LOHN, JOSHUA A	
ART UNIT	**PAPER NUMBER**
2114	
MAIL DATE	**DELIVERY MODE**
05/03/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/045,307	**Applicant(s)** DEWITT ET AL.	
	Examiner Joshua A. Lohn	**Art Unit** 2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 December 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *2,3,5,6,8 and 9* is/are allowed.

6)☒ Claim(s) *1,4,7 and 10-33* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 January 2002* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Response to Arguments

Applicant's arguments filed 12/15/2006 have been fully considered but they are not persuasive.

With respect to applicant's arguments involving claim 1, that Moughani fails to disclose the receiving step of claim 1, the examiner respectfully disagrees. Claim 1 recites that the receiving includes at least a portion of the executable code from an application has been loaded into a memory block. In contrast, Moughani discloses that information is received to indicate that a memory area may be accessed. However, the "valid" indication of Moughani is directly related to an indication that the code is executable since the "valid" value indicates that a block is defined by an application and executable (Moughani, col. 4, lines 8-9). Applicant further argues that Moughani doesn't disclose receiving at a tracing function, only at an operating system or user level. Examiner disagrees since the Moughani reference discloses the receiving by the system or user is all part of the greater memory tracing function, and discloses receiving by a tracing function (Moughani, col. 3, lines 51-53).

With respect to applicant's arguments involving claim 1, that Moughani fails to disclose the altering step of claim 1, the examiner respectfully disagrees. Applicant argues that the supervisor is responsible for the altering, and not a tracing function. However the examiner disagrees since the supervisor's setting action is still a part of the greater tracing function (Moughani, figure 6, col. 3, lines 51-53).

With respect to applicant's arguments involving claims 4 and 7, the examiner respectfully disagrees for the same reasons as those stated above with respect to claim 1.

With respect to applicant's arguments involving claim 10, that IBM fails to disclose the allocating step in the manner recited in claim 10, the examiner respectfully disagrees. While it is true that IBM teaches against a "memory buffer", the physical I/O storage of IBM is functionally equivalent to the output buffer since the limitation, as it is currently claimed, merely requires an output buffer acting as a physical storage for the trace data. The IBM reference discloses the I/O storage, which is functionally equivalent to a broadly claimed output buffer.

With respect to applicant's arguments involving claim 11, the examiner respectfully disagrees for the same reasons as those stated above with respect to claim 10.

With respect to applicant's further arguments involving claim 11, that the IBM reference teaches an algorithm, not a routine, and especially not a register, the examiner respectfully disagrees. The algorithm of the reference controls memory and interrupt routines that are used to write memory to a specified address in the input/output space. The configuring of this algorithm is functionally equivalent to having a translation register in that it is used to control the memory operation, which collects the trace data and translates it for an output buffer in the I/O space.

With respect to applicant's arguments involving claims 13 and 15, the examiner respectfully disagrees for the same reasons as those stated above with respect to claim 11.

With respect to applicant's arguments involving claim 16, that Levine does not teach the second writing step in the manner recited in claim 16, the examiner respectfully disagrees. The claim recites a buffer to hold instruction resolution information, and applicant argues that the register of Levine is not the same as a buffer, and that a buffer as known in the art holds data until it can be written elsewhere. However, the Levine register also holds data, which is importing to the tracing, and is thus held in a functionally equivalent manner to a buffer for storing as part of the trace. Further, applicant Levine teaches a change in mapping, which takes less space than storage in a buffer. However this argument in irrelevant to the limitation as currently claimed, where there is no necessity disclosed for control of the storage space in a buffer. The storage must only store operating-system-defined memory allocation information. The examiner feels that the disclosure of Levine fully clearly teaches the memory allocation information in the address mapping change information storage (Levine, col. 9, lines 25-43).

With respect to applicant's arguments involving claims 22 and dependents 17-21 and 23-27, the examiner respectfully disagrees for the same reasons as those stated above with respect to claim 16.

With respect to applicant's arguments involving the 103 rejection of claims 28, and dependents 29-33 that Levine and Tanenbaum fail to teach the second writing step, the examiner respectfully disagrees. The examiner feels that Levine discloses this second writing step as is shown above with respect to claim 16.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1, 4, and 7 are rejected under 35 U.S.C. 102(b) as being anticipated by Moughani et al., United States Patent, 5,970,246, published October 19, 1999.

As per claim 1, Moughani discloses a method for reducing interrupts while tracing an application in a data processing system, the method comprising: receiving at a tracing function an indication that at least a portion of executable code from an application has been loaded into a memory block (Moughani, col. 4, lines 5-34, where the user accesses memory that must be valid and verified, meaning the memory has loaded executable code) prior to execution of the portion of executable code (Moughani, col. 4, lines 35-40, where the access is made with the intent to execute, but the trace bit causes the access to first generate a trace); and altering by the tracing function at least one operating-system-defined memory access protection parameter to allow read access to the memory block (Moughani, col. 3, line 51, through col. 4, line 4, where the altering is done by the setting of the trace bit).

As per claim 4, Moughani discloses an apparatus for reducing interrupts while tracing an application in a data processing system, the apparatus comprising: means for receiving at a tracing function an indication that at least a portion of executable code from an application has been loaded into a memory block (Moughani, col. 4, lines 5-34, where the user accesses memory that must be valid and verified, meaning the memory has loaded executable code) prior to execution of the portion of executable code (Moughani, col. 4, lines 35-40, where the access is made with the intent to execute, but the trace bit causes the access to first generate a trace); and means for altering by the tracing function at least one operating-system-defined memory access protection parameter to allow read access to the memory block (Moughani, col. 3, line 51, through col. 4, line 4, where the altering is done by the setting of the trace bit).

As per claim 7, Moughani discloses a computer program product in a computer-readable medium for use in a data processing system for reducing interrupts while tracing an application (Moughani, col. 3, line 51, through col. 4, line 41, where the supervisor mode and the various bit settings are controlled by a computer program), the computer program product comprising: instructions for receiving at a tracing function an indication that at least a portion of executable code from an application has been loaded into a memory block (Moughani, col. 4, lines 5-34, where the user accesses memory that must be valid and verified, meaning the memory has loaded executable code) prior to execution of the portion of executable code (Moughani, col. 4, lines 35-40, where the access is made with the intent to execute, but the trace bit causes the access to first generate a trace); and instructions for altering by the tracing function at least one operating-system-defined memory access protection parameter to allow read access to the memory block (Moughani, col. 3, line 51, through col. 4, line 4, where the altering is done by the setting of the trace bit).

Claims 10-15 are rejected under 35 U.S.C. 102(b) as being anticipated by "Processor Single Step Trace Facility Enhancements", IBM Technical Disclosure Bulletin NN961231, published December 1, 1996.

As per claim 10, TDB NN961231 discloses a method for reducing interrupts while tracing an application in a data processing system, the method comprising: initiating execution of tracing software (TDB NN961231, first paragraph); allocating a data output buffer in physical memory, wherein the data output buffer holds output data from the tracing software (TDB NN961231, second paragraph, where the writing to the I/O space is an access to physical memory); and writing output data to the data output buffer by the tracing software using physical memory addressing (TDB NN961231, second paragraph, where the I/O space is directly addressed).

As per claim 11, TDB NN961231 discloses a method for reducing interrupts while tracing an application in a data processing system, the method comprising: initiating execution of tracing software (TDB NN961231, first paragraph); allocating a data output buffer, wherein the data output buffer holds output data from the tracing software (TDB NN961231, second paragraph, where an area of the I/O space is initialized); and configuring a translation register in a processor of the data processing system for the data output buffer (TDB NN961231, second paragraph, where the interrupt routine functions as a translation register by providing access to the specified memory address for the trace data that is transmitted, and translated to this memory space).

As per claim 12, TDB NN961231 discloses an apparatus for reducing interrupts while tracing an application in a data processing system, the apparatus comprising: means for initiating execution of tracing software (TDB NN961231, first paragraph), wherein a data output buffer holds output data from the tracing software (TDB NN961231, second paragraph, where an area of the I/O space is initialized); and means for writing output data to the data output buffer by the tracing software using physical memory addressing (TDB NN961231, second paragraph, where the I/O space is directly addressed).

As per claim 13, TDB NN961231 discloses an apparatus for reducing interrupts while tracing an application in a data processing system, the apparatus comprising: means for initiating execution of tracing software (TDB NN961231, first paragraph), wherein a data output buffer holds output data from the tracing software (TDB NN961231, second paragraph, where an area of the I/O space is initialized); and means for configuring a translation register in a processor of the data processing system for the data output buffer (TDB NN961231, second paragraph, where the interrupt routine functions as a translation register by providing access to the specified memory address for the trace data that is transmitted, and translated to this memory space).

As per claim 14, TDB NN961231 discloses a computer program product in a computer-readable medium (TDB NN961231, second and third paragraphs, where the instruction/data tracing is an software algorithm that determines how to proceed in the tracing of the system) for use in a data processing system for reducing interrupts while tracing an application, the computer program product comprising: instructions for initiating execution of tracing software (TDB NN961231, first paragraph), wherein a data output buffer holds output data from the tracing software (TDB NN961231, second paragraph, where an area of the I/O space is initialized); and instructions for writing output data to the data output buffer by the tracing software using physical memory addressing (TDB NN961231, second paragraph, where the I/O space is directly addressed).

As per claim 15, TDB NN961231 discloses a computer program product in a computer-readable medium (TDB NN961231, second and third paragraphs, where the instruction/data tracing is an software algorithm that determines how to proceed in the tracing of the system) for use in a data processing system for reducing interrupts while tracing an application, the computer program product comprising: instructions for initiating execution of tracing software (TDB NN961231, first paragraph), wherein a data output buffer holds output data from the tracing software (TDB NN961231, second paragraph, where an area of the I/O space is initialized); and instructions for configuring a translation register in a processor of the data processing system for the data output buffer (TDB NN961231, second paragraph, where the interrupt routine functions as a translation register by providing access to the specified memory address for the trace data that is transmitted, and translated to this memory space).

Claims 16-27 and 34-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Levine et al., United States Patent number 5,446,876, published August 29, 1995.

As per claim 16, Levine discloses a method for reducing interrupts while tracing an application in a data processing system, the method comprising: receiving an indication of an instruction to be traced, wherein the instruction is associated with an instruction address (Levine, col. 4, lines 51-59); in response to receiving the indication of the instruction to be traced, retrieving the instruction address; writing the instruction address to a trace output buffer in memory (Levine, col. 9, lines 10-11); and writing instruction resolution information to a trace output buffer, wherein the instruction resolution information comprises operating-system-defined memory allocation information or generated application code (Levine, col. 8, lines 25-43)

As per claim 17, Levine discloses receiving an indication of a change to memory allocation information for an application, wherein the step of writing operating-system-defined memory allocation information is performed in response to receiving the indication of the change to memory allocation information for the application (Levine, col. 8, lines 25-43).

As per claim 18, Levine discloses reconciling the instruction address with the operating-system-defined memory allocation information to determine a location of the instruction in an application file or module (Levine, col. 8, lines 32-35).

As per claim 19, Levine discloses retrieving a copy of the instruction from an application file or module in relation to the instruction address (Levine, col. 8, lines 1-24)

As per claim 20, Levine discloses reconciling the instruction address with the generated application code to determine a location of the instruction within the generated application code (Levine, col. 5, lines 1-22, where the generated transactions may alter memory mappings that must then be reconciled).

As per claim 21, Levine discloses retrieving a copy of the instruction from the generated application code in relation to the instruction address (Levine, col. 5, lines 1-22, where the instruction is received based upon generated code that alters address mappings).

As per claim 22, Levine discloses an apparatus for reducing interrupts while tracing an application in a data processing system, the apparatus comprising: means for receiving an indication of an instruction to be traced, wherein the instruction is associated with an instruction address (Levine, col. 4, lines 51-59); means for retrieving the instruction address in response to receiving the indication of the instruction to be traced; means for writing the instruction address to a trace output buffer in memory (Levine, col. 9, lines 10-11); and means for writing instruction resolution information to a trace output buffer, wherein the instruction resolution information comprises operating-system-defined memory allocation information or generated application code (Levine, col. 8, lines 25-43).

As per claim 23, Levine discloses means for receiving an indication of a change to memory allocation information for an application, wherein the step of writing operating-system-defined memory allocation information is performed in response to receiving the indication of the change to memory allocation information for the application (Levine, col. 8, lines 25-43).

As per claim 24, Levine discloses means for reconciling the instruction address with the operating-system-defined memory allocation information to determine a location of the instruction in an application file or module (Levine, col. 8, lines 32-35).

As per claim 25, Levine discloses means for retrieving a copy of the instruction from an application file or module in relation to the instruction address (Levine, col. 8, lines 1-24).

As per claim 26, Levine discloses means for reconciling the instruction address with the generated application code to determine a location of the instruction within the generated application code (Levine, col. 5, lines 1-22, where the generated transactions may alter memory mappings that must then be reconciled).

As per claim 27, Levine discloses means for retrieving a copy of the instruction from the generated application code in relation to the instruction address (Levine, col. 5, lines 1-22, where the instruction is received based upon generated code that alters address mappings).

As per claim 34, Levine discloses a method for reducing interrupts while tracing an application in a data processing system, the method comprising: receiving an indication of an instruction to be traced, wherein the instruction is associated with an instruction address (Levine, col. 4, lines 55-57); storing the instruction address (Levine, col. 4, lines 57-59); getting a previously stored instruction address (Levine, col. 4, lines 59-62); retrieving a previously executed instruction using the previously stored instruction address (Levine, col. 4, lines 64-66); and writing the retrieved instruction to a trace output buffer (Levine, col. 4, lines 67-68).

As per claim 35, Levine discloses that a processor in the data processing system supports variable length instructions (Levine, col. 5, lines 6-11).

As per claim 36, Levine discloses retrieving a branch-from address; and retrieving a set of previously executed instructions using the previously stored instruction address and the branch-from address (Levine, col. 4, lines 64-68).

As per claim 37, Levine discloses an apparatus for reducing interrupts while tracing an application in a data processing system, the apparatus comprising: means for receiving an indication of an instruction to be traced, wherein the instruction is associated with an instruction address (Levine, col. 4, lines 55-57); means for storing the instruction address (Levine, col. 4, lines 57-59); means for getting a previously stored instruction address (Levine, col. 4, lines 59-62); means for retrieving a previously executed instruction using the previously stored instruction address (Levine, col. 4, lines 64-66); and means for writing the retrieved instruction to a trace output buffer (Levine, col. 4, lines 67-68).

As per claim 38, Levine discloses a processor in the data processing system supports variable length instructions (Levine, col. 5, lines 6-11).

As per claim 39, Levine discloses means for retrieving a branch-from address; and means for retrieving a set of previously executed instructions using the previously stored instruction address and the branch-from address (Levine, col. 4, lines 64-68).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 28-33 and 40-42 rejected under 35 U.S.C. 103(a) as being unpatentable over Levine, in view of Tanenbaum, Structured Computer Organization, Third Edition, published 1990.

As per claim 28, Levine discloses a method for tracing comprising: means for receiving an indication of an instruction to be traced, wherein the instruction is associated with an instruction address (Levine, col. 4, lines 51-59); means for retrieving the instruction address in response to receiving the indication of the instruction to be traced; means for writing the instruction address to a trace output buffer in memory (Levine, col. 9, lines 10-11); and means for writing instruction resolution information to a trace output buffer, wherein the instruction resolution information comprises operating-system-defined memory allocation information or generated application code (Levine, col. 8, lines 25-43). Levine fails to disclose this method being implemented using a computer program product in a computer-readable medium.

Tanenbaum discloses that hardware and software are logically equivalent (Tanenbaum, page 11).

It would have been obvious to one skilled in the art at the time of the invention to implement the invention of Levine in a software form based upon the teachings of Tanenbaum.

This would have bee obvious because Tanenbaum discloses that many factors associated with computers can be altered in a beneficial way by implementing a system in software rather than hardware (Tanenbaum, page 11). The fact that the systems are logically equivalent shows that it would be obvious to change between implementations if any of the possible improvements to cost, speed, reliability, and frequency are to be realized.

As per claim 29, Levine and Tanenbaum disclose means for receiving an indication of a change to memory allocation information for an application, wherein the step of writing operating-system-defined memory allocation information is performed in response to receiving the indication of the change to memory allocation information for the application (Levine, col. 8, lines 25-43).

As per claim 30, Levine and Tanenbaum disclose means for reconciling the instruction address with the operating-system-defined memory allocation information to determine a location of the instruction in an application file or module (Levine, col. 8, lines 32-35).

As per claim 31, Levine and Tanenbaum disclose means for retrieving a copy of the instruction from an application file or module in relation to the instruction address (Levine, col. 8, lines 1-24).

As per claim 32, Levine and Tanenbaum disclose means for reconciling the instruction address with the generated application code to determine a location of the instruction within the generated application code (Levine, col. 5, lines 1-22, where the generated transactions may alter memory mappings that must then be reconciled.

As per claim 33, Levine and Tanenbaum disclose means for retrieving a copy of the instruction from the generated application code in relation to the instruction address (Levine, col.

5, lines 1-22, where the instruction is received based upon generated code that alters address mappings).

As per claim 40, Levine discloses a method comprising: instructions for receiving an indication of an instruction to be traced, wherein the instruction is associated with an instruction address (Levine, col. 4, lines 55-57); instructions for storing the instruction address (Levine, col. 4, lines 57-59); instructions for getting a previously stored instruction address (Levine, col. 4, lines 59-62); instructions for retrieving a previously executed instruction using the previously stored instruction address (Levine, col. 4, lines 64-66); and instructions for writing the retrieved instruction to a trace output buffer (Levine, col. 4, lines 67-68). Levine fails to disclose this method being implemented using a computer program product including instructions in a computer-readable medium.

Tanenbaum discloses that hardware and software are logically equivalent (Tanenbaum, page 11).

It would have been obvious to one skilled in the art at the time of the invention to implement the invention of Levine in a software form based upon the teachings of Tanenbaum.

This would have bee obvious because Tanenbaum discloses that many factors associated with computers can be altered in a beneficial way by implementing a system in software rather than hardware (Tanenbaum, page 11). The fact that the systems are logically equivalent shows that it would be obvious to change between implementations if any of the possible improvements to cost, speed, reliability, and frequency are to be realized.

As per claim 41, Levine and Tanenbaum disclose that a processor in the data processing system supports variable length instructions (Levine, col. 5, lines 6-11).

As per claim 42, Levine and Tanenbaum disclose instructions for retrieving a branch-from address; and instructions for retrieving a set of previously executed instructions using the previously stored instruction address and the branch-from address (Levine, col. 4, lines 64-68).

Allowable Subject Matter

Claims 2-3, 5-6, and 8-9 are allowable.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joshua A. Lohn whose telephone number is (571) 272-3661. The examiner can normally be reached on M-F 8-4.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Scott Baderman can be reached on (571) 272-3644. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JAL

SCOTT BADERMAN
SUPERVISORY PATENT EXAMINER